the Availability Digest

Do You Know Where Your Train Is?

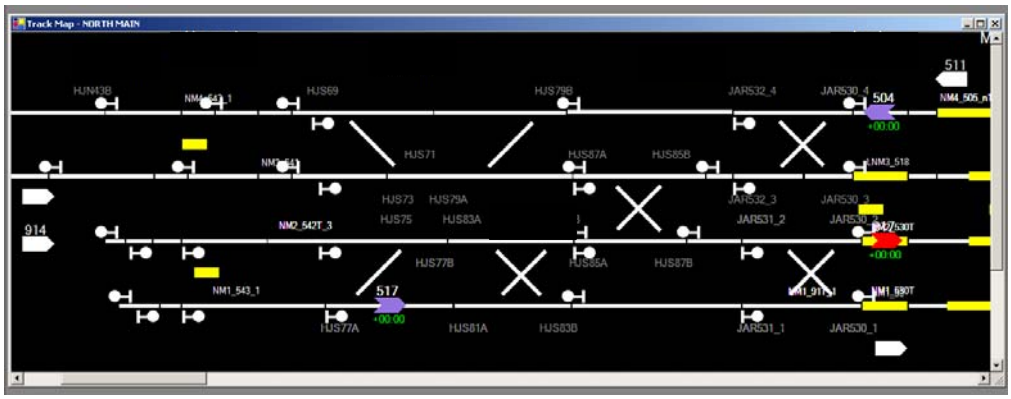
October 2006

A public transit authority for a large metropolitan city had been monitoring its train locations with an aging train tracking and identification system. It became imperative to upgrade this system; and to do so, the transit authority turned to QuicTrak, a product from QEI Inc., of Springfield, New Jersey (www.qeiinc.com). QEI is a major supplier of SCADA (Supervisory Control and Data Acquisition) systems to the utility and transportation industries and had supplied the transit authority with its SCADA system for monitoring and controlling traction and signal power for the railroad.

We first describe the QuicTrak functions and architecture and then explain the extensive fault-monitoring and failover provisions of QuicTrak – the subject in which we are most interested.

QuicTrak

QuicTrak provides train monitoring and identification services for railroad systems via a pictorial display of the track system for the train controllers. As trains move through the system, train icons are displayed appropriately on a track map. Each train is identified by its run number, which is derived when the train enters the system according to the published transit schedules. The current schedule adherence time is also shown for each train.



A QuicTrak Track Map

Tracking information is received from the field via a QEI TDMS Plus SCADA system. This information includes such data as track segment occupancies, switch positions, signal conditions, gate positions, bridge positions, and so on. The sequence of track occupancies as a train moves through the system is used to calculate the position of that train and to display an icon representing it on track maps displayed on the operator consoles.

Train position information is also used to post the anticipated arrivals of trains on display boards located at each station.

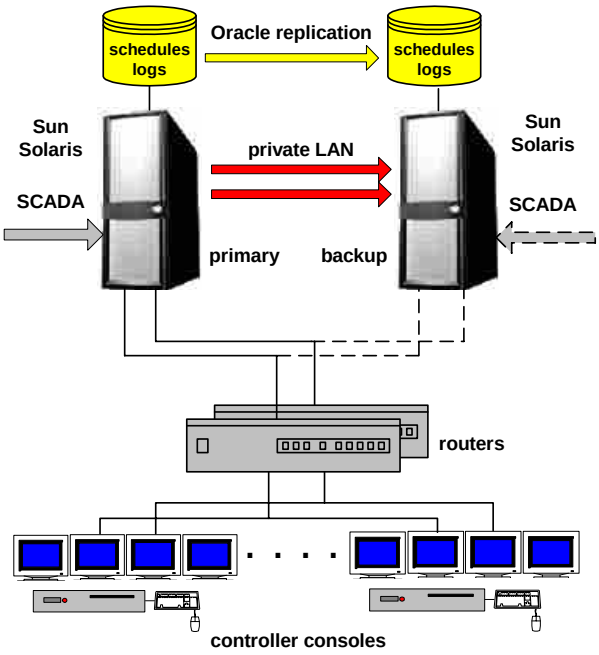
The System Configuration

The QuicTrak application is written in Java so that QuicTrak is platform-independent. Though QEI generally uses VAX OpenVMS hardware for its SCADA systems, the transit authority wanted to use Sun servers. The redundant QuicTrak system is therefore configured with a pair of Sun V880 900 megahertz servers, each with four gigabytes of memory and an Oracle 9 database. The servers run the Sun Solaris operating system. Separate databases resident on each subsystem were selected rather than a common redundant database system since the transit authority may want to geographically distribute the two nodes in the future for disaster tolerance purposes.

The servers are interconnected with a high speed LAN; and the databases, which hold the train schedules and the logs of indications and controller actions, are kept in synchronism with Oracle 9's data replication facility.

Originally, it was intended that the two Sun servers would be synchronized by simply replicating their databases to each other. However, it turned out that the Oracle replicator was much too slow for this. Its replication latency was measured as several seconds. Therefore, it was decided to use Oracle replication only for replicating the schedules and log files.

To keep the two system states in synchronism, a private redundant dual LAN is provided between the two systems. System state is replicated over this link as trains move along.



QuicTrak Configuration

Though the system is configured as an active/active system with memory-to-memory replication of system state, it cannot be used in a purely active/active manner since incoming indications must be processed in exact sequence. Otherwise, a train could be seen as jumping back and forth between consecutive track segments as it traveled along. Therefore, the system is run in "sizzling hot standby" mode in which, like a fully functional active/active system, users are immediately switched to the surviving node in the event of the failure of the primary node.

The controller consoles are connected to each system via a router. The router routes all traffic to and from the consoles to the primary node. If the primary node fails, the users are immediately switched to the backup node by the router. We describe failure monitoring and failover procedures later.

Each of the consoles is four-headed so that a track map, which tends to be linear in nature, can be spread across all four consoles if desired.

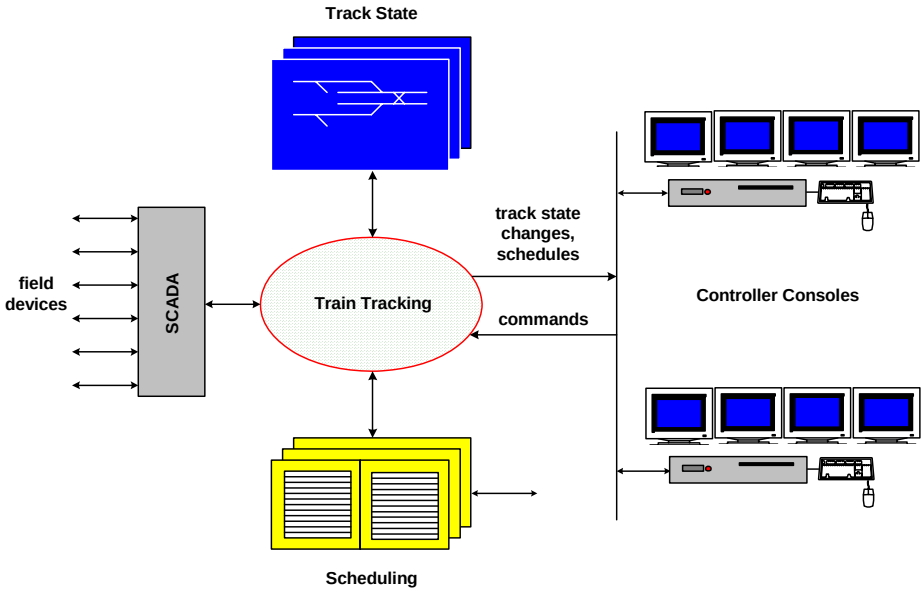
Though the QuicTrak system can support dual LANs, the transit authority decided to use only one LAN to connect the controller consoles to the system. It based this decision on its experience with its corporate LAN.

The Software Architecture

The primary application in QuicTrak is the Train Tracking module. It is responsible for all of the logic required to determine where the trains are located and to identify each train against the transit schedule. It works with four other modules – the SCADA subsystem, the current Track State, the Scheduling subsystem, and the consoles.

The SCADA subsystem communicates with the field devices over communication links provided for this purpose. It receives from these devices track occupancies, switch positions, signal states, gate and bridge positions, and so on. It is these indications that provide Train Tracking the information to properly determine the position of all trains and to display the state of all other devices.

The SCADA system also sends to the field the controls that have been commanded by the controllers from their consoles.



QuicTrak Software Architecture

The Track State is a memory-resident map of the entire train system. It contains the state of every field device, the current position of all trains, their identification and schedule adherence times, and so on. By applying the incoming indications to the current Track State, Train Tracking can determine train movement and can detect alarm conditions.

The Scheduling subsystem maintains the transit schedules. These can be modified by controller actions as trains begin to run late during the day. A common need is to spread out trains if one is running late. The measure of service in a transit system is not whether the trains are running on time but rather by how much time a passenger must wait for the next train.

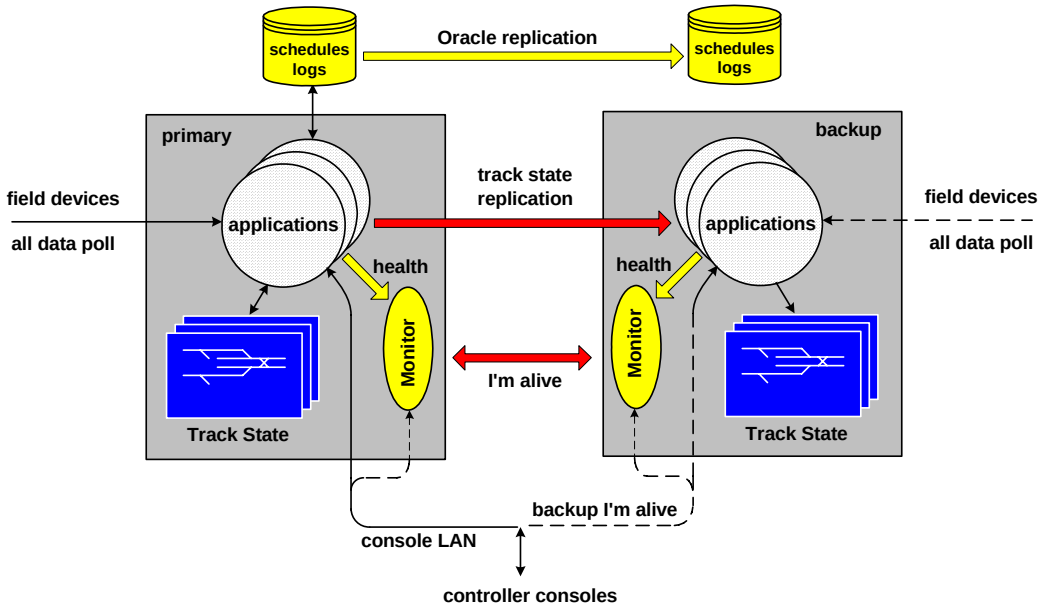
Train tracking uses the schedules to identify trains as they enter the system. It also updates the current status of each train run in the schedule. This status is used to post next train times at display boards in the stations.

Each of the consoles contains the entire Track State so that it does not have to be continually querying the server to get state information. Whenever Train Tracking changes the state of the system, the Track State changes are broadcast to all of the consoles. The consoles are also used by the controllers to generate commands to the field to throw switches or to change signal states.

A controller can also call up schedules and display them so that he can see what movements are expected and can resolve any train identification confusion that may have occurred.

Failure Monitoring and Failover

Since we are interested primarily in availability, the monitoring and failover capabilities of QuicTrak are of most interest to us. This is implemented via a Monitor that resides in both systems.



Monitoring and Failover

System Health

Each application (for instance, Train Tracking, Scheduling, SCADA) periodically reports its health to its Monitor. The reporting time is configurable but is typically one second. Health is reported on a scale of 1 to 10. If an application is fully operational, its health is 10. If it determines that it is compromised, it will report a lower health rating - for instance, if it finds that its queue of work is becoming too long or if it is seeing high error rates on a communication line. If an application does not report within a configurable number of reporting cycles, the Monitor assumes that it has failed and gives that application a health score of 0.

Based on the reported health of all of its applications, the Monitor calculates an overall health metric for its system.

Health reporting is active in both the primary and the backup systems. Each Monitor reports its system's health and that of its applications to the other Monitor via "I'm alive" messages.

Failure Determination

If the backup Monitor determines that the health of the primary system has fallen to an unacceptable level, and if its health is better than that of the primary, it will command a failover. This determination is based not only on the overall system health metrics but also on those of the individual applications. For instance, if all applications in the primary but one are reporting perfect health and that one application is down (a health of zero) and is a critical application, the backup Monitor will command a failover.

Likewise, if the primary Monitor determines that the backup system is down, it will report this for maintenance action. It is as important to keep the backup healthy as it is to keep the primary healthy. Otherwise, a primary failure could force a failover to an inoperable backup system.

Failover

As described earlier, the backup's state has been kept synchronized with the primary system's state via memory-to-memory replication. Therefore, failover is effected by simply switching users to the backup system, which now becomes the primary system. This switchover is implemented via the *gratuitous ARP* facility within TCP/IP, as described next.

ARP is the Address Resolution Protocol used by routers to maintain in their cache a mapping of IP addresses to device interface addresses on the subnets to which they are connected. This mapping allows the router to determine where to send a datagram that it has received. Routers will periodically send their routing tables to neighboring routers via ARP. In this way, routing tables are self-discovering. They always keep up-to-date on the current network topology.

If a router does not have a destination on its subnet to which to forward an incoming message, it sends out an *ARP request* over the subnet and asks for the interface servicing that IP address. The appropriate device will respond with its interface address. This will be used by every router on the subnetwork to update its cache.

An important characteristic of a routing table is that it is updated by the IP/interface address pairs contained in each message that is sent over the subnets to which the router is connected, whether that message is a datagram or an ARP request. Every message contains the sending IP address and its associated device interface address on the connecting subnet as well as that pair of addresses for the destination

A gratuitous ARP is a form of ARP request in which the sender asks for its own address. As described above, the ARP request will carry the sender's IP address and interface address. As a consequence, each router on the subnet will update its routing table and will associate that IP address with the device interface address of the sender. Therefore, sending a gratuitous ARP allows the sender to seize that IP address.

Using this facility, the backup system takes over control by sending out a gratuitous ARP. All console messages will thereafter be routed to the backup The backup will then

- notify the old primary system to shut down,
- activate Train Tracking so that it can communicate with the SCADA subsystem,
- allow local applications to write to the backup's disk-based schedules and log files,
- reverse the direction of replication so that its schedules, log files, and Track State changes are replicated to the other system, and

- refresh the console's Track State caches so that they are in synchronism with the system Track State.

At this point, the old backup has become the new primary system. Failover has been done in less than one second.

Loss of the Interprocessor Connection

Key to the monitoring process is the communication between the Monitors on each system. They communicate over a high-speed dedicated LAN. To ensure that this connection is always available, dual LANs are provided. Furthermore, should QuicTrak lose both LANs, the console LAN is also used as a backup monitoring connection. If communication is lost over all three of these connections (a highly unlikely occurrence), the backup will assume that the primary has failed and will initiate failover.

Tug-of-War

A tug-of-war can occur if both systems think that they should be primary. However, in the case of QuicTrak, following a failover, the old primary system has been commanded to go down. Therefore, it cannot initiate another failover until corrective action has been taken and until it is once again put back online.

The one case in which a tug-of-war can occur is if there is a failure of all three communication links used for intermonitor communication. However, with the triple redundancy provided, this is highly unlikely. Moreover, if the Monitors are unable to communicate over the console LAN, it is likely that the entire system is down anyway.

Lost Transactions

It is possible that one or more Track State changes still in the replication queue could be lost following a failover. Therefore, one of the first tasks for the new primary is to command its SCADA system to do an *all data poll* of its field devices to get the latest states of these devices. This can take several seconds.

Also, it is good practice for each controller to check following a failover that his last command has, in fact, been executed. If there is a question, the controller should repeat that command.

Production Cutover

The transit authority began using QuicTrak in June, 2006. Interestingly, cutover was delayed by a problem mentioned earlier. We noted that, though properly configured, this system could not be run in true active/active mode because all indications had to be processed in time-sequence order. However, this problem occurred anyway but for a totally different and unanticipated reason.

It turned out that the signaling systems used in the field were quite old. Not only were they somewhat unreliable (a problem which is partially corrected by QEI's SCADA system and by QuicTrak), but the polling cycles were quite long, in the order of several seconds. Since this was a local transit system, many of the instrumented track segments were quite short (in the order of 50 feet). Therefore, in certain areas, multiple track occupancy changes for the same stretch of track could come in on one poll. Since these changes were not time-tagged, they were processed in arbitrary order. This made it look like the train was jumping back and forth between track segments.

The problem was solved in part by speeding up the polling process and in part by using only selected track segments in such sections. Following this correction, the system worked properly and was put into production.

Steve Dalyai, President of QEI, can be contacted at sdalyai@qeiinc.com.